
서버 프로그램 구현

정기 수행 평가 06



Contents

1 프로젝트 생성

- 프로젝트 구조
- 구조 상세

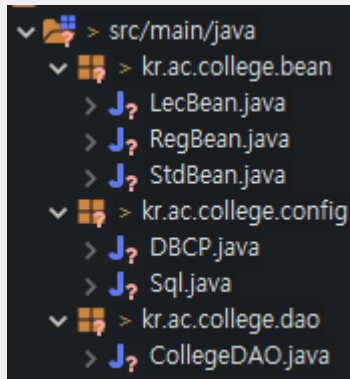
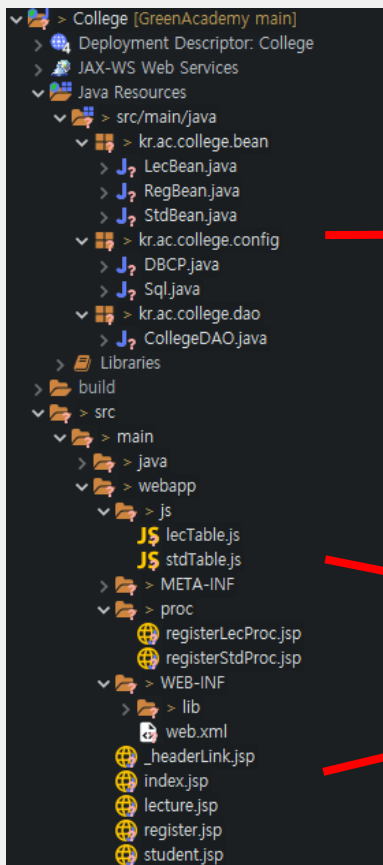
2 화면 구현 및 동작 화면

- lecture 구현
- student 구현
- register 구현

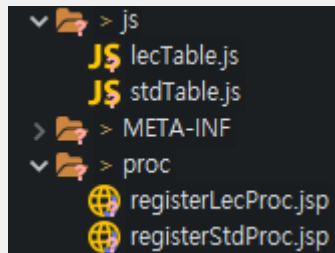
1

프로젝트 생성

| 프로젝트 구조

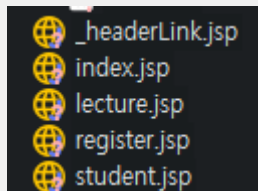


DB 관련 클래스들



기능 구현

모듈화 된 파일들



메인 화면

1 프로젝트 생성

| 구조 상세

```
//Java1_College
<Resource
  name="dbcp_java1_college"
  auth="Container"
  type="javax.sql.DataSource"
  driverClassName="com.mysql.cj.jdbc.Driver"
  url="jdbc:mysql://127.0.0.1:3306/java1_college"
  username="root"
  password="1234"
  maxTotal="8"
  maxIdle="8"
  maxWaitMillis="3000"
/>
```

톰캣 서버 context.xml에 커넥션 풀 생성

```
1 <%@ page pageEncoding="utf-8"%>
2
3 <a href="/College/lecture.jsp">강좌관리</a>
4 <a href="/College/register.jsp">수강관리</a>
5 <a href="/College/student.jsp">학생관리</a>
```

```
<body>
  <h3>강좌관리</h3>
  <%@ include file="./_headerLink.jsp" %>
```

_headerLink.jsp 파일과 본문 импорт

```
public class RegBean {
  private String regStdNo;
  private String regLecNo;
  private String regMidScore;
  private String regFinalScore;
  private String regTotalScore;

  private String regGrade;

  //추가필드
  private String regStdName;
  private String regLecName;
```

```
public int getStdYear() {
  return stdYear;
}
public void setStdYear(int stdYear) {
  this.stdYear = stdYear;
}
public void setStdYear(String stdYear) {
  this.stdYear = Integer.parseInt(stdYear);
}
```

Bean 클래스, 추가 필드 및 생성자로 확장성 증가

```
function init(lectures){
  $(function(){
    $('.lecList').empty();

    $('.lecList').append("<h4>강좌현황</h4>");
    $('.lecList').append("<button type='button' class='btnCreateLecTable'>등록</button>");
    let tableTag = "<table border='1'>"
    tableTag += "<tr>";
    tableTag += "<th>번호</th>";
    tableTag += "<th>강좌명</th>";
    tableTag += "<th>환경</th>";
    tableTag += "<th>시간</th>";
    tableTag += "<th>강의장</th>";
    tableTag += "</tr>";
    for(let lb in lectures){
      tableTag += "<tr>";
    }
    $('.lecList').append(tableTag);
  });
}
```

JS 파일

1

프로젝트 생성

| 구조 상세

```
public class CollegeDAO {
    private static CollegeDAO instance = new CollegeDAO();
    public static CollegeDAO getInstance() {
        return instance;
    }
    private CollegeDAO() {};

    //CRUD 구현
    //Lecture
    public List<LecBean> selectLectures() {}

    public int insertLecture(LecBean lb) {}
    //

    //Register
    public List<RegBean> selectRegisters() {}

    public int insertRegister(RegBean rb) {}

    //

    //Student
    public List<StdBean> selectStudents() {}

    public int insertStudent(StdBean sb) {}
    //
}
```

```
public class Sql {
    //lecture
    public static final String SELECT_LECTURES = "SELECT * FROM `lecture`";
    public static final String INSERT_LECTURE = "INSERT INTO `lecture` VALUES (?, ?, ?, ?, ?)";

    //register
    public static final String SELECT_REGISTERS = "SELECT a.*, b.`stdName`, c.`lecName` "
        + "FROM `register` AS a JOIN `student` AS b "
        + "ON a.`regStdNo` = b.`stdNo` "
        + "JOIN `lecture` AS c "
        + "ON a.`regLecNo` = c.`lecNo`";
    public static final String INSERT_REGISTER = "INSERT INTO `register` SET "
        + "`regStdNo`=?, `regLecNo`=?";

    //student
    public static final String SELECT_STUDENTS = "SELECT * FROM `student`";
    public static final String INSERT_STUDENT = "INSERT INTO `student` VALUES (?, ?, ?, ?, ?)";
}
```

```
public class DBCP {}

    private static DataSource ds = null;

    public static Connection getConnection() throws NamingException, SQLException {
        if(ds == null) {
            ds = (DataSource) new InitialContext().lookup("java:comp/env/dbcp_java1_college");
        }
        return ds.getConnection();
    }
}
```

DAO, Sql, Bean으로 구현해 성능을 높이고 유지보수에 용이한 코드 구현

2 화면 구현 및 동작 화면

| lecture 화면 구현

```
38<body>
39  <h3>강좌관리</h3>
40  <%@ include file="../headerLink.jsp" %>
41
42  <div class="lecList">
43    <h4>강좌현황</h4>
44    <button type="button" class="btnCreateLecTable">등록</button>
45    <table border="1">
46      <tr>
47        <th>번호</th>
48        <th>강좌명</th>
49        <th>학점</th>
50        <th>시간</th>
51        <th>강의장</th>
52      </tr>
53      <% for(LecBean lecture : lectures) {%>
54        <tr>
55          <td><%= lecture.getLecNo() %></td>
56          <td><%= lecture.getLecName() %></td>
57          <td><%= lecture.getLecCredit() %></td>
58          <td><%= lecture.getLecTime() %></td>
59          <td><%= lecture.getLecClass() %></td>
60        </tr>
61      <% } %>
62    </table>
63  </div>
64  <div class="lecTable"></div>
65 </body>
66 </html>
```

student.jsp(html부)

```
1 <%@ page pageEncoding="utf-8"%>
2
3 <a href="/College/lecture.jsp">강좌관리</a>
4 <a href="/College/register.jsp">수강관리</a>
5 <a href="/College/student.jsp">학생관리</a>
```

학생관리

[강좌관리](#) [수강관리](#) [학생관리](#)

학생목록

등록

학번	이름	휴대폰	학년	주소
20201016	김유신	010-1234-1001	3	null
20201126	김춘추	010-1234-1002	3	경상남도 경주시
20210216	장보고	010-1234-1003	2	전라남도 완도시
20210326	강감찬	010-1234-1004	2	서울시 영등포구
20220416	이순신	010-1234-1005	1	부산시 부산진구
20220521	송상현	010-1234-1006	1	부산시 동래구

_headerLink.jsp 및 실제 화면

2

| lecture 기능 구현

```

50<%
6    CollegeDAO dao = CollegeDAO.getInstance();
7    //강좌목록 가져오기
8    List<LecBean> lectures = dao.selectLectures();
9 %>
10 <!DOCTYPE html>
11<html>
12< head>
13     <meta charset="UTF-8">
14     <title>College:lecture</title>
15     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
16     <script src="/College/js/lecTable.js"></script>
17</head>
18     $(function){
19         //강좌등록 테이블 생성
20         $(document).on('click', '.btnCreateLecTable', function(e){
21             e.preventDefault();
22             createLecTable();
23         });
24         //강좌등록 테이블 제거
25         $(document).on('click', '.btnCloseLecTable', function(e){
26             e.preventDefault();
27             closeLecTable();
28         });
29         //강좌 등록
30         $(document).on('click', 'input[type=submit]', function(e){
31             e.preventDefault();
32             registerLecTable();
33             createLecTable();
34         });
35     };
36 </script>
37 </head>

```

```

1 <%@page import="com.google.gson.JsonObject"%>
2 <%@page import="kr.ac.college.bean.LecBean"%>
3 <%@page import="kr.ac.college.dao.CollegeDAO"%>
4 <%@ page contentType="application/json;charset=UTF-8" pageEn
5 <%
6     request.setCharacterEncoding("utf-8");
7     String lecNo      = request.getParameter("lecNo");
8     String lecName     = request.getParameter("lecName");
9     String lecCredit   = request.getParameter("lecCredit");
10    String lecTime      = request.getParameter("lecTime");
11    String lecClass     = request.getParameter("lecClass");
12
13    CollegeDAO dao = CollegeDAO.getInstance();
14    LecBean lb = new LecBean();
15    int result = 0;
16
17    lb.setLecNo(lecNo);
18    lb.setLecName(lecName);
19    lb.setLecCredit(lecCredit);
20    lb.setLecTime(lecTime);
21    lb.setLecClass(lecClass);
22
23    result = dao.insertLecture(lb);
24
25    JsonObject json = new JsonObject();
26    json.addProperty("result", result);
27    String jsonData = json.toString();
28
29    out.print(jsonData);
30 >%

```

```

28@function createLecTable(){
29
30    $(function(){
31        $('<table>').empty();
32
33        $('<table>').append("<tr><td>강의등록</td>");
34        $('<table>').append("<button type='button' class='btnCloseLecTable'>닫기</button>");
35        let tableTag = "<table border='1'>"
36        tableTag += "<tr>";
37        tableTag += "<td>번호</td>";
38        tableTag += "<td><input type='text' name='lecNo'></td>";
39        tableTag += "</tr>";
40        tableTag += "<tr>";
41        tableTag += "<td>강의명</td>";
42        tableTag += "<td><input type='text' name='lecName'></td>";
43        tableTag += "</tr>";
44        tableTag += "<tr>";
45        tableTag += "<td>학금</td>";
46        tableTag += "<td><input type='text' name='lecCredit'></td>";
47        tableTag += "</tr>";
48        tableTag += "<tr>";
49        tableTag += "<td>시간</td>";
50        tableTag += "<td><input type='text' name='lecTime'></td>";
51        tableTag += "</tr>";
52        tableTag += "<tr>";
53        tableTag += "<td>강의장</td>";
54        tableTag += "<td><input type='text' name='lecClass'></td>";
55        tableTag += "</tr>";
56        tableTag += "<tr>";
57        tableTag += "<td colspan='2' align='right'><input type='submit' value='추가'></td>";
58        tableTag += "</tr>";
59        tableTag += "</table>";
60
61        $('<table>').append(tableTag);
62    });
63
64 }
65
66@function closeLecTable(){
67    $('<table>').empty();
68 }

```

Lecture.jsp (기능할당)

registerLecProc.jsp

Ajax 요청에 응답

lecTable.js (테이블 생성 및 삭제버튼)

2 화면 구현 및 동작 화면

| lecture 기능 구현

```
66 function closeLecTable(){
67     $('#lecTable').empty();
68 }
69
70 function registerLecTable(){}
71
72 $(function(){
73
74     //데이터 가져오기
75     let lecNo = $('#input[name=lecNo]').val();
76     let lecName = $('#input[name=lecName]').val();
77     let lecCredit = $('#input[name=lecCredit]').val();
78     let lecTime = $('#input[name=lecTime]').val();
79     let lecClass = $('#input[name=lecClass]').val();
80
81     //JSON 데이터 생성
82     let jsonData = {
83         "lecNo":lecNo,
84         "lecName":lecName,
85         "lecCredit":lecCredit,
86         "lecTime":lecTime,
87         "lecClass":lecClass
88     }
89
90     //데이터 전송
91     $.ajax({
92         url: '/College/proc/registerLecProc.jsp',
93         type: 'POST',
94         data: jsonData,
95         datatype: 'json',
96         success:function(data){
97             if(data.result == 1){
98                 alert('추가 완료!');
99             }else{
100                 alert('입력 정보를 확인해주세요.');
101             }
102         }
103     });
104
105 });
```

lecTable.js
(등록 버튼)

```
//lecture
public static final String SELECT_LECTURES = "SELECT * FROM `lecture`";
public static final String INSERT_LECTURE = "INSERT INTO `lecture` VALUES (?, ?, ?, ?, ?)";
```

```
public List<LecBean> selectLectures() {
    List<LecBean> lectures = null;
    try {
        lectures = new ArrayList<>();
        Connection conn = DBCP.getConnection();
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(Sql.SELECT_LECTURES);
        while(rs.next()) {
            LecBean lb = new LecBean();
            lb.setLecNo(rs.getInt(1));
            lb.setLecName(rs.getString(2));
            lb.setLecCredit(rs.getInt(3));
            lb.setLecTime(rs.getInt(4));
            lb.setLecClass(rs.getString(5));

            lectures.add(lb);
        }
        rs.close();
        stmt.close();
        conn.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return lectures;
}
```

```
public int insertLecture(LecBean lb) {
    int result = 0;
    try {
        Connection conn = DBCP.getConnection();
        PreparedStatement psmt =
            conn.prepareStatement(Sql.INSERT_LECTURE);
        psmt.setInt(1, lb.getLecNo());
        psmt.setString(2, lb.getLecName());
        psmt.setInt(3, lb.getLecCredit());
        psmt.setInt(4, lb.getLecTime());
        psmt.setString(5, lb.getLecClass());

        result = psmt.executeUpdate();
        System.out.println(result);
        psmt.close();
        conn.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return result;
}
```

CollegeDAO.java, Sql.java
Lecture 기능 구현

2 화면 구현 및 동작 화면

| lecture 화면 동작

강좌관리

[강좌관리](#) [수강관리](#) [학생관리](#)

강좌현황

등록					
번호	강좌명	학점	시간	강의장	
159	인지행동심리학	3	40	본304	
167	운영체제론	3	25	본B05	
234	무역영문	3	20	본201	
239	세법개론	3	40	본204	
248	빅데이터 개론	3	20	본B01	
253	컴퓨팅사고와 코딩	2	10	본B02	
349	사회복지 마케팅	2	50	본301	

강좌등록

닫기

번호	<input type="text"/>
강좌명	<input type="text"/>
학점	<input type="text"/>
시간	<input type="text"/>
강의장	<input type="text"/>
<input type="button" value="추가"/>	

강좌관리

[강좌관리](#) [수강관리](#) [학생관리](#)

강좌현황

등록					
번호	강좌명	학점	시간	강의장	
159	인지행동심리학	3	40	본304	
167	운영체제론	3	25	본B05	
234	무역영문	3	20	본201	
239	세법개론	3	40	본204	
248	빅데이터 개론	3	20	본B01	
253	컴퓨팅사고와 코딩	2	10	본B02	
349	사회복지 마케팅	2	50	본301	

강좌등록

닫기

번호	<input type="text" value="123"/>
강좌명	<input type="text" value="test"/>
학점	<input type="text" value="1"/>
시간	<input type="text" value="23"/>
강의장	<input type="text" value="본123"/>
<input type="button" value="추가"/>	

localhost:8080 내용:

추가 완료!

강좌관리

[강좌관리](#) [수강관리](#) [학생관리](#)

강좌현황

등록				
번호	강좌명	학점	시간	강의장
123	test	1	23	본123
159	인지행동심리학	3	40	본304
167	운영체제론	3	25	본B05
234	무역영문	3	20	본201
239	세법개론	3	40	본204
248	빅데이터 개론	3	20	본B01
253	컴퓨팅사고와 코딩	2	10	본B02
349	사회복지 마케팅	2	50	본301

실제 동작 화면

2 화면 구현 및 동작 화면

| student 화면 구현

```
40<body>
41  <h3>학생관리</h3>
42  <%@ include file="../_headerLink.jsp" %>
43
44  <div class="stdList">
45    <h4>학생목록</h4>
46    <button type="button" class="btnCreateStdTable">등록</button>
47    <table border="1">
48      <tr>
49        <th>학번</th>
50        <th>이름</th>
51        <th>휴대폰</th>
52        <th>학년</th>
53        <th>주소</th>
54      </tr>
55      <% for(StdBean student : students) {%>
56      <tr>
57        <td><%= student.getStdNo() %></td>
58        <td><%= student.getStdName() %></td>
59        <td><%= student.getStdHp() %></td>
60        <td><%= student.getStdYear() %></td>
61        <td><%= student.getStdAddress() %></td>
62      </tr>
63      <% } %>
64    </table>
65  </div>
66  <div class="stdTable"></div>
67 </body>
68 </html>
```

student.jsp(html부)

```
1 <%@ page pageEncoding="utf-8"%>
2
3 <a href="/College/lecture.jsp">강좌관리</a>
4 <a href="/College/register.jsp">수강관리</a>
5 <a href="/College/student.jsp">학생관리</a>
```

강좌관리

[강좌관리](#) [수강관리](#) [학생관리](#)

강좌현황

등록					
번호	강좌명	학점	시간	강의장	
159	인지행동심리학	3	40	본304	
167	운영체제론	3	25	본B05	
234	무역영문	3	20	본201	
239	세법개론	3	40	본204	
248	빅데이터 개론	3	20	본B01	
253	컴퓨팅사고와 코딩	2	10	본B02	
349	사회복지 마케팅	2	50	본301	

_headerLink.jsp 및 실제 화면

2 화면 구현 및 동작 화면

| student 기능 구현

```
5<%
6    CollegeDAO dao = CollegeDAO.getInstance();
7    //학생등록 가져오기
8    List<StdBean> students = dao.selectStudents();
9 %>
10 <!DOCTYPE html>
11<html>
12<head>
13<meta charset="UTF-8">
14<title>College::Student</title>
15<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
16<script src="/College/js/stdTable.js"></script>
17<script>
18    $(function(){
19        //강좌등록 테이블 생성
20        $(document).on('click', '.btnCreateStdTable', function(e){
21            e.preventDefault();
22            createStdTable();
23        });
24        //강좌등록 테이블 제거
25        $(document).on('click', '.btnCloseStdTable', function(e){
26            e.preventDefault();
27            closeStdTable();
28        });
29        //강좌 등록
30        $(document).on('click', 'input[type=submit]', function(e){
31            e.preventDefault();
32            registerStdTable();
33            createLecTable();
34        });
35    });
36</script>
37</head>
```

student.jsp (기능할당)

```
1<%%page import="kr.ac.college.bean.StdBean"%>
2<%%page import="com.google.gson.JsonObject"%>
3<%%page import="kr.ac.college.dao.CollegeDAO"%>
4<%% page contentType="application/json;charset=UTF-8" pageEncoding="UTF-8"%>
5<%
6    request.setCharacterEncoding("utf-8");
7    String stdNo = request.getParameter("stdNo");
8    String stdName = request.getParameter("stdName");
9    String stdHp = request.getParameter("stdHp");
10    String stdYear = request.getParameter("stdYear");
11    String stdAddress = request.getParameter("stdAddress");
12
13    CollegeDAO dao = CollegeDAO.getInstance();
14    StdBean sb = new StdBean();
15    int result = 0;
16
17    sb.setStdNo(stdNo);
18    sb.setStdName(stdName);
19    sb.setStdHp(stdHp);
20    sb.setStdYear(stdYear);
21    sb.setStdAddress(stdAddress);
22
23    result = dao.insertStudent(sb);
24    JsonObject json = new JsonObject();
25    json.addProperty("result", result);
26    String jsonData = json.toString();
27
28    out.print(jsonData);
29
30 %>
```

registerStdProc.jsp
Ajax 요청에 응답

```
6function createStdTable(){
7
8    $(function(){
9        $('<table>').empty();
10
11        $('<table>').append("<tr><td>강좌등록</td></tr>");
12        $('<table>').append("<tr><td><input type='button' value='닫기' /></td></tr>");
13        let tableTag = "<table border='1'>";
14        tableTag += "<tr>";
15        tableTag += "<td>강좌등록</td>";
16        tableTag += "<td><input type='button' value='닫기' /></td>";
17        tableTag += "</tr>";
18        tableTag += "<tr>";
19        tableTag += "<td>이름</td>";
20        tableTag += "<td><input type='text' name='stdName' /></td>";
21        tableTag += "</tr>";
22        tableTag += "<tr>";
23        tableTag += "<td>휴대폰</td>";
24        tableTag += "<td><input type='text' name='stdHp' /></td>";
25        tableTag += "</tr>";
26        tableTag += "<tr>";
27        tableTag += "<td>학년</td>";
28        tableTag += "<td><input type='text' name='stdYear' /></td>";
29        tableTag += "</tr>";
30        tableTag += "<tr>";
31        tableTag += "<td>주소</td>";
32        tableTag += "<td><input type='text' name='stdAddress' /></td>";
33        tableTag += "</tr>";
34        tableTag += "<tr>";
35        tableTag += "<td colspan='2' align='right'><input type='submit' value='추가' /></td>";
36        tableTag += "</tr>";
37        tableTag += "</table>";
38
39        $('<table>').append(tableTag);
40    });
41
42 }
43
44function closeStdTable(){
45    $('<table>').empty();
46 }
```

stdTable.js
(테이블 생성 및 삭제버튼)

2 화면 구현 및 동작 화면

| student 기능 구현

```
48 function registerStdTable() {
49   $(function(){
50     //데이터 가져오기
51     let stdNo = $('input[name=stdNo]').val();
52     let stdName = $('input[name=stdName]').val();
53     let stdHp = $('input[name=stdHp]').val();
54     let stdYear = $('input[name=stdYear]').val();
55     let stdAddress = $('input[name=stdAddress]').val();
56
57     //JSON 데이터 생성
58     let jsonData = {
59       "stdNo":stdNo,
60       "stdName":stdName,
61       "stdHp":stdHp,
62       "stdYear":stdYear,
63       "stdAddress":stdAddress
64     };
65
66     //데이터 전송
67     $.ajax({
68       url: '/College/proc/registerStdProc.jsp',
69       type: 'POST',
70       data: jsonData,
71       datatype: 'json',
72       success:function(data){
73         if(data.result == 1){
74           alert('추가 완료!');
75         }else{
76           alert('입력 정보를 확인해주세요.');
77         }
78       }
79     });
80   });
81 }
```

stdTable.js
(등록 버튼)

```
//student
public static final String SELECT_STUDENTS = "SELECT * FROM `student`";
public static final String INSERT_STUDENT = "INSERT INTO `student` VALUES (?, ?, ?, ?, ?)";
```

```
//Student
public List<StdBean> selectStudents() {
    List<StdBean> students = null;
    try {
        students = new ArrayList<>();
        Connection conn = DBCP.getConnection();
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(Sql.SELECT_STUDENTS);
        while(rs.next()) {
            StdBean sb = new StdBean();
            sb.setStdNo(rs.getString(1));
            sb.setStdName(rs.getString(2));
            sb.setStdHp(rs.getString(3));
            sb.setStdYear(rs.getInt(4));
            sb.setStdAddress(rs.getString(5));

            students.add(sb);
        }
        rs.close();
        stmt.close();
        conn.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return students;
}
```

```
public int insertStudent(StdBean sb) {
    int result = 0;
    try {
        Connection conn = DBCP.getConnection();
        PreparedStatement pstmt =
            conn.prepareStatement(Sql.INSERT_STUDENT);
        pstmt.setString(1, sb.getStdNo());
        pstmt.setString(2, sb.getStdName());
        pstmt.setString(3, sb.getStdHp());
        pstmt.setInt(4, sb.getStdYear());
        pstmt.setString(5, sb.getStdAddress());

        result = pstmt.executeUpdate();
        System.out.println(result);
        pstmt.close();
        conn.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return result;
}
```

CollegeDAO.java, Sql.java
Student 기능 구현

2 화면 구현 및 동작 화면

| lecture 화면 동작

학생관리

[강좌관리](#) [수강관리](#) [학생관리](#)

학생목록

등록				
학번	이름	휴대폰	학년	주소
20201016	김유신	010-1234-1001	3	null
20201126	김준추	010-1234-1002	3	경상남도 경주시
20210216	장보고	010-1234-1003	2	전라남도 완도시
20210326	강감찬	010-1234-1004	2	서울시 영등포구
20220416	이순신	010-1234-1005	1	부산시 부산진구
20220521	송상현	010-1234-1006	1	부산시 동래구

강좌등록

닫기

학번	
이름	
휴대폰	
학년	
주소	
<div>추가</div>	

학생관리

[강좌관리](#) [수강관리](#) [학생관리](#)

학생목록

등록				
학번	이름	휴대폰	학년	주소
20201016	김유신	010-1234-1001	3	null
20201126	김준추	010-1234-1002	3	경상남도 경주시
20210216	장보고	010-1234-1003	2	전라남도 완도시
20210326	강감찬	010-1234-1004	2	서울시 영등포구
20220416	이순신	010-1234-1005	1	부산시 부산진구
20220521	송상현	010-1234-1006	1	부산시 동래구

강좌등록

닫기

학번	2022test
이름	test
휴대폰	010-1234-test
학년	0
주소	test
<div>추가</div>	

localhost:8080 내용:

추가 완료!

학생관리

[강좌관리](#) [수강관리](#) [학생관리](#)

학생목록

등록				
학번	이름	휴대폰	학년	주소
20201016	김유신	010-1234-1001	3	null
20201126	김준추	010-1234-1002	3	경상남도 경주시
20210216	장보고	010-1234-1003	2	전라남도 완도시
20210326	강감찬	010-1234-1004	2	서울시 영등포구
20220416	이순신	010-1234-1005	1	부산시 부산진구
20220521	송상현	010-1234-1006	1	부산시 동래구
2022test	test	010-1234-test	0	test

실제 동작 화면

2 화면 구현 및 동작 화면

| register 구현

```
//register
public static final String SELECT_REGISTERS = "SELECT a.*, b.`stdName`, c.`lecName` "
      + "FROM `register` AS a JOIN `student` AS b "
      + "ON a.`regStdNo` = b.`stdNo` "
      + "JOIN `lecture` AS c "
      + "ON a.`regLecNo` = c.`lecNo`";

//where 추가
public static final String INSERT_REGISTER = "INSERT INTO `register` SET "
      + "`regStdNo`=?, `regLecNo`=?";
```

Sql.java

수강관리

[강좌관리](#) [수강관리](#) [학생관리](#)

수강현황

등록

학번	이름	강좌명	강좌코드	중간시험	기말시험	총점시험	등급시험
20201126	김춘추	무역영문	234	36	42	78	C
20201016	김유신	빅데이터 개론	248	24	62	86	B
20201016	김유신	컴퓨팅사고와 코딩	253	28	40	68	D
20201126	김춘추	세법개론	239	37	57	94	A
20210216	장보고	사회복지 마케팅	349	28	68	96	A
20210326	강감찬	사회복지 마케팅	349	16	65	81	B
20201016	김유신	운영체제론	167	18	38	56	F
20220416	이순신	사회복지 마케팅	349	25	58	83	B

```
public class RegBean {
    private String regStdNo;
    private String regLecNo;
    private String regMidScore;
    private String regFinalScore;
    private String regTotalScore;

    private String regGrade;

    //추가필드
    private String regStdName;
    private String regLecName;
```

RegBean.java

정기 수행 평가

06.서버 프로그램 구현 College



Thank you!

Kim Dong Geun
From GreenAcademy
2022.11.08